REMARKS

The Examiner's Action mailed on January 7, 2009, has been received and its contents carefully considered. Reconsideration of the final rejections presented therein is requested for at least the following reasons.

Claims 1 and 9 are the independent claims, and claims 1-14 remain pending in the application. For at least the following reasons, it is submitted that this application is in condition for allowance.

Claims 1-14 were rejected under 35 U.S.C. §103(a) as being obvious over *Alexander* (U.S. 6,553,029 B1) in view of *Tursich* (U.S. 6,671,828 B1). This rejection is respectfully traversed.

Claim 1 recites:

> A multiple port single chip Ethernet switch comprising at least the following component parts:
>
> a physical layer entity (PHY) including a plurality of ports;
>
> an address table for being written to and read out information to operate the plurality of ports;
>
> a switch for switching the Ethernet switch to a daisy chain test mode; and
>
> an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip;
>
> wherein said component parts of said Ethernet switch are formed on said single chip.

Claim 9 recites:

> A daisy chain test for a single chip Ethernet switch integrated with a physical layer entity including a plurality of ports, the switch having an address table for being written to and read out information to operate the plurality of ports, the test comprising the steps of:
>
> connecting each of the plurality of ports to a respective passive loop-back device;
>
> selecting a start transmission port and a stop receiving port from the plurality of ports;
>
> . supplying a test packet to the start transmission port; and
>
> proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function; and
>
> determining a test result by verifying a last received packet at the stop receiving port.

In *Alexander*, address resolution unit **10** extracts the source and destination MAC addresses from each Ethernet packet, and looks up the addresses in MAC address look-up table **12** to determine if associated contexts exist for the source and destination MAC addresses. If they exist, address resolution unit **10** then presents the packet data along with the source and destination MAC address contexts to embedded CPU **14** for processing by packet forwarding firmware routine **20**. See col. 4, lines 26-35:

> Address resolution unit **10** thus extracts the source and destination MAC addresses from each Ethernet packet, looks up the addresses in MAC address look-up table **12** to determine if associated contexts exist for the source and destination MAC addresses, and also retrieves the contexts from table **12** if they exist. Address resolution unit **10** then presents the packet data along with the source and destination MAC address contexts (if found) to embedded CPU **14** for processing by packet forwarding firmware routine **20** (FIG. 2, block **28**).

If they do not exist, address table creation ("learning function") firmware routine **16** is invoked to update MAC address look-up table **12**. See col. 4, lines 39-44:

> Address table creation ("learning function") firmware routine **16** is invoked when address resolution unit **10** determines that a particular source MAC address found within a packet is not present in address look-up table **12**; in which case table **12** is updated accordingly.

Specifically, the learning function is used to update the MAC address look-up table **12**, and the learning function is performed by the firmware routine **16**, which is not included in the address resolution unit **10**. The address resolution unit **10** of *Alexander* is used to perform an address look-up process, and never performs the learning function. See col. 4, lines 17-20:

> Address resolution unit **10** accepts multiple streams of packets (one stream from each physical link: FIG. 2, block 24) and performs an address look-up process on each packet using MAC address look-up table **12** (FIG. 2, block **26**).

However, in the present application, the packet source address learning process is used to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip, and the packet source address learning process is performed by the source address learning engine which is included in the address resolution control logic (i.e., the packet source address learning process is performed by the address resolution control logic).

Thus, the address resolution unit **10** of *Alexander* does not correspond to the address resolution control logic of the present application, and the learning function of *Alexander* also differs from the packet source address learning process of the present application.

Hence, *Alexander* fails to teach or suggest either "an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip" as recited in Claim 1, or "proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function" as recited in Claim 9.

Moreover, *Alexander* fails to disclose a daisy chain test mode, which is admitted in the Office Action. Therefore, *Alexander* necessarily fails to teach or suggest "a switch for switching the Ethernet switch to a daisy chain test mode" as recited in Claim 1, or "determining a test result by verifying a last received packet at the stop receiving port" as recited in Claim 9.

In *Tursich*, the test system screens packets entering the daisy-chain based on the destination addresses in the packets to control the traffic in a daisy-chain of protocol analyzers, and therefore a cost-effective test system is created. See col. 1, line 65 to col. 2, line 5:

> The invention solves the above problem with a cost-effective test system that controls the traffic in a daisy-chain of protocol analyzers. The test system screens packets entering the daisy-chain based on the destination addresses in the packets. Advantageously, the system is implemented with simple control processing that does not add significant cost or complexity to the packet network or to the protocol analyzers.

More specifically, *Tursich* only discloses that protocol analyzers **110**, **120** and **130** daisy-chained together, but never discloses a daisy chain test mode. See FIG. 1 and col. 2, lines 53-54:

> Protocol analyzers **110**, **120**, and **130** are daisy-chained together.

Tursich therefore also fails to teach or suggest "a switch for switching the Ethernet switch to a daisy chain test mode" as recited in Claim 1, or "determining a test result by verifying a last received packet at the stop receiving port" as recited in Claim 9.

Moreover, in *Tursich*, the control system receives the packet from the first interface and <u>either deletes the packet or transfers the packet to the second interface</u> based on a destination address in the packet, and the second interface transfers the packet to another protocol analyzer. See the Abstract of the Disclosure and col. 2, line 59 to col. 3, line 3:

> A protocol analyzer for a test system has a control system coupled to a first interface and a second interface. The first interface receives a packet from a packet network. The control system receives the packet from the first interface and either deletes the packet or transfers the packet to the second interface based on a destination address in the packet. The second interface transfers the packet to another protocol analyzer. The destination address could be a MAC address.
>
> Protocol analyzers **110**, **120**, and **130** monitor traffic on lines-under-test **151** to generate and transfer test packets to test computer **140**. Test computer **140** processes the test packets to generate test results regarding target communication device **150** for network operators. Test computer **140** also transfers test packets to protocol analyzers **110**, **120**, and **130** to control testing. Protocol analyzer **130** receives the test packets from test computer **140**. Protocol analyzer **130** also receives other packets carried by packet network **152**. Protocol analyzer **130** processes some packets internally, transfers some packets to protocol analyzers **110** and **120**, and deletes the other packets.

Tursich thus fails to disclose a packet source address learning process for delivering a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip.

Hence, *Tursich* also fails to teach or suggest either "an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip" as recited in Claim 1, or "proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function" as recited in Claim 9.

Therefore, *Alexander* and *Tursich*, whether taken separately or in combination, fail to teach all the features of independent claims 1 and 9.

Further, there is no reason to combine *Alexander* with *Tursich*, because *Alexander* fails to teach a need for testing.

Furthermore, in *Tursich*, test system **100** comprising protocol analyzers **110**, **120**, and **130** is used to monitor traffic on lines-under-test **151** to generate and transfer test packets to test computer **140**, and test computer **140** processes the test pockets to generate test results regarding target communication device **150** for network operators. See *Tursich*, col. 2, lines 59-63:

> Protocol analyzers **110**, **120**, and **130** monitor traffic on lines-under-test **151** to generate and transfer test packets to test computer **140**. Test computer **140** processes the test packets to generate test results regarding target communication device **150** for network operators.

More specifically, the test packets are transferred to the test computer **140** rather than delivered through the plurality of ports progressively from a start transmission port to a stop receiving port, and the test packets are transferred between different devices (i.e., protocol analyzers **110**, **120**, and **130** and test computer **140**) to control testing rather than between the plurality of ports in a single device to test the single device. See col. 2, lines 63-65:

> Test computer **140** also transfers test packets to protocol analyzers **110**, **120**, and **130** to control testing.

Consequently, neither *Alexander* nor *Tursich*, whether taken separately or in combination, teaches or suggests all the features recited in claim 1 or claim 9, and therefore claims 1 and 9 are allowable, together with claims 2-8 and 10-14 that depend respectively therefrom.

It is submitted that this application is in condition for allowance. Such action and the passing of this case to issue are requested.

Should the Examiner feel that a conference would help to expedite the prosecution of this application, the Examiner is hereby invited to contact the undersigned counsel to arrange for such an interview.

Should the remittance be accidentally missing or insufficient, the Commissioner is hereby authorized to charge the fee to our Deposit Account No. 18-0002, and advise us accordingly.

Respectfully submitted,

Alun Palmer

March 31, 2009
Date

Alun L. Palmer – Reg. No. 47,838
RABIN & BERDO, PC – Cust. No. 23995
Facsimile: 202-408-0924
Telephone: 202-371-8976

ALP/